NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

APPLICABILITY OF DEEP-LEARNING TECHNOLOGY FOR RELATIVE OBJECT-BASED NAVIGATION

by

Wee Leong Lai

September 2017

Thesis Advisor:	Oleg A. Yakimenko
Second Reader:	Fotis A. Papoulias

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE APPLICABILITY OF DEEP-LEARNING TECHNOLOGY FOR RELATIVE OBJECT-BASED NAVIGATION			**5. FUNDING NUMBERS**
6. AUTHOR(S) Wee Leong Lai			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approve for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

In a GPS-denied environment, one of the possible selections for navigating an unmanned ground vehicle (UGV) is through real-time visual odometry. To navigate in such an environment, the UGV needs to be able to detect, identify, and relate the static and dynamic objects such as trucks, motorbikes, and pedestrians in the on-board camera field of view. Therefore, object recognition becomes crucial in navigating UGVs. However, object recognition is known to be one of the challenges in the field of computer vision. Current analytic video software inadequately utilizes heuristics like size, shape, and direction to determine whether a detected object is a human, a vehicle, or an animal. This thesis explores another approach, the deep-learning technique, which makes use of neural networks based on vast collections of training data images. This thesis follows a systems engineering approach in analyzing the need and suggesting a solution. It shows how to create and train the aforementioned networks using just three objects: a chair, a table, and a car. A Pioneer UGV equipped with the corresponding sensors is then used to test the developed algorithms. The preliminary analysis conducted in this thesis shows good potential for using the deep-learning technique on future UGVs.

14. SUBJECT TERMS unmanned ground vehicle, navigation, computer vision, deep learning, object recognition			**15. NUMBER OF PAGES** 65
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approve for public release. Distribution is unlimited.

APPLICABILITY OF DEEP-LEARNING TECHNOLOGY FOR RELATIVE OBJECT-BASED NAVIGATION

Wee Leong Lai
Civilian, ST Electronics (Info-Comm Systems) Pte Ltd.
BEng, Nanyang Technological University, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2017

Approved by: Oleg A. Yakimenko, Ph.D.
Thesis Advisor

Fotis A. Papoulias, Ph.D.
Second Reader

Ronald E. Giachetti, Ph.D.
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In a GPS-denied environment, one of the possible selections for navigating an unmanned ground vehicle (UGV) is through real-time visual odometry. To navigate in such an environment, the UGV needs to be able to detect, identify, and relate the static and dynamic objects such as trucks, motorbikes, and pedestrians in the on-board camera field of view. Therefore, object recognition becomes crucial in navigating UGVs. However, object recognition is known to be one of the challenges in the field of computer vision. Current analytic video software inadequately utilizes heuristics like size, shape, and direction to determine whether a detected object is a human, a vehicle, or an animal. This thesis explores another approach, the deep-learning technique, which makes use of neural networks based on vast collections of training data images. This thesis follows a systems engineering approach in analyzing the need and suggesting a solution. It shows how to create and train the aforementioned networks using just three objects: a chair, a table, and a car. A Pioneer UGV equipped with the corresponding sensors is then used to test the developed algorithms. The preliminary analysis conducted in this thesis shows good potential for using the deep-learning technique on future UGVs.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	artificial intelligence
CCD	charge-coupled device
CNN	convolutional neural network
CPU	central processing unit
DARPA	Defense Science and Technology Agency
DOD	Department of Defense
EMC	electromagnetic compatibility
EMI	electromagnetic interference
EO	electro optical
FOS	family of system
FOV	field of view
GPS	global positioning system
GUIDE	GUI development environment
HD	high definition
HIS	human system interface
JPEG	joint photographic experts group
KPPs	key performance parameters
OS	operating system
RGB	red, green and blue
ROS	robot operating system
SE	systems engineering
UGV	unmanned ground vehicle
U.S.	United States
USB	universal serial bus
VGA	video graphics array

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This thesis explored the applicability of deep-learning technology for relative object-based navigation in an urban environment with a degraded GPS signal. In such a critical mission, using just GPS, static sensors and map data as the navigation tool is not sufficient. There is a need to involve additional sensors including cameras. The optical sensor is a popular choice as it can collect tremendous amounts of information such as live video feeds, video recording, capture static image, video analytics and object recognition. This information aids the UGV and operators in understanding the environment and planning/ adjusting the course of actions.

This thesis follows systems engineering procedures to develop a deep-learning based system and test it in a series of representative test cases. The deep-learning technology explored in this thesis is a subset of machine learning. It utilizes convolutional neural network (CNN) to learn the image features automatically from large repository of training image dataset. There are three techniques that can be successfully deployed for CNN on image classification and this thesis used one of them, the transfer learning approach. This approach happens to be more practical to use with an existing pretrained model such as Alexnet to improve the image classification accuracy due to small training data.

This research utilized a Pioneer UGV with an on-board day camera to conduct the test the developed deep-learning algorithm. The test case consisted of three different types of test scenarios with three different types of training images datasets. The three test scenarios are identification of a single object which consist of both indoor and outdoor environment test, identification of multiple homogeneous objects and identification of multiple heterogeneous objects. Besides that, three different types of training images dataset were setup for each of the test scenarios to compare the system accuracy. The three different types of training images dataset are 20 training images, that original 20 plus new 20 training images, and 39 training images from an earlier dataset plus 1 image of the actual scene. Ten test cycles run were conducted for each test scenario to validate that the system was able to provide consistently good results.

Based on the conducted research, the results shows that the accuracy of the deep-learning based system improve with the increase of training images in the dataset. In addition, the test with 39 training images from earlier dataset plus 1 image of the actual scene has obtained the best overall best results. The results demonstrate that there is a lot of potential in this research for future work.

ACKNOWLEDGMENTS

I wish to express my gratitude and appreciation to those who have generously helped me in completing this thesis, especially to Professor Oleg Yakimenko and Professor Fotis Papoulias. This thesis would not have been successful without their encouragement, constructive suggestions, and precious guidance through the research period. It has been a great joy and honor to work with them. In addition, I would like to pay a special tribute to Professor Brian Bingham and Professor Barbara Berlitz, who have both put in a lot of time and effort throughout the research and the report.

I am grateful to my family and wife, Evelyn, who has provided me moral and emotional support throughout the entire course of my study at the Naval Postgraduate School.

A very special thanks goes out to ST Electronics (Info-Comm Systems) Pte Ltd. for putting their faith in me and providing me this opportunity.

Last but not least, I would like to express my gratitude to the Naval Postgraduate School for providing excellent research facilities and financial support for this project.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Unmanned ground vehicles (UGVs) have made major leaps in the use of military technology for modern urban warfare. Technology advancement has enhanced their capability to provide aid and to complete tasks that are deemed too risky or mundane for the solders in war (Snider and Simon 2016). Those tasks include intelligence, surveillance, and reconnaissance (ISR), explosive ordinance disposal (EOD), and search and rescue operation. The UGV can handle those tasks in place of a human to minimize the exposure of danger to soldiers (Hanlon 2005). Figure 1 displays the UGV's family of systems (FOS) that supports the U.S. military combating units such as Airforce, Army, Navy, and others (Winnefeld and Kendall 2011, 22).

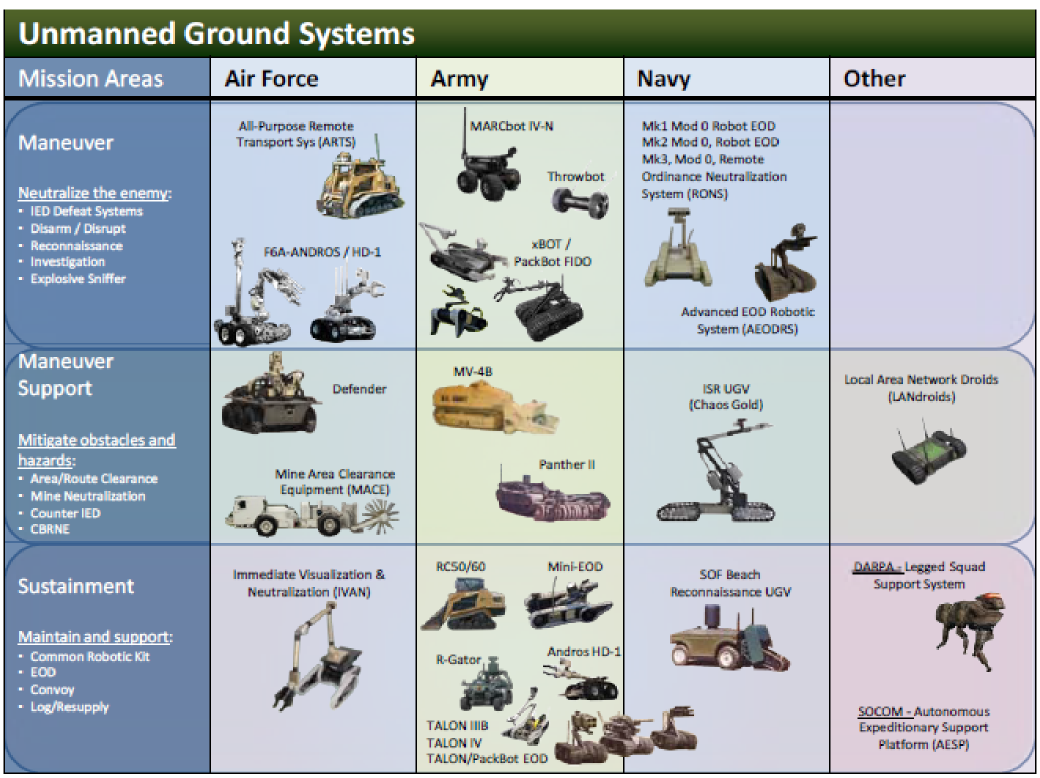

Figure 1. UGVs FOS. Source: Winnefeld and Kendall (2011, 22).

From Winnefeld and Kendall (2011), the U.S. military deployed up to 8,000 UGVs for the Operation Enduring Freedom and the Operation Iraqi Freedom. These UGVs had completed more than 125,000 missions such as EOD, object identification, and others, as recorded in September 2010 (James A and Kendall 2011, 22). Further, the UGVs have effectively assisted U.S. military to detect and destroy more than 11,000 EODs (Winnefeld and Kendall 2011, 23). With the global military technology landscape changing at an unprecedented pace, there is a need to constantly update its defense capabilities such as technological advancement and tactical changes. Therefore, it is important to study and analyze the applicability of upcoming technological trends to stay ahead of potential vulnerability.

1. Technological Advancement

Although different sensors have been deployed for UGV application, which has led to a varied spectrum of solutions. In the last three decades, there is vast research into visual navigation for mobile robots. Vision system is small which can be easily installed on space limited mobile robots. It also provides situation awareness of the event with either live video streaming or by image capture (Bonin-Font et al. 2008). To reduce the workload of the operators, video analytics was implemented in the vision system. The video analytics primarily assist in tracking objects and making heuristic guesses of an object's position.

By contrast, deep learning technology makes use of convolutional neural network that learn from large training data to achieve highly accurate in object classification. The increase in the accuracy of object detection and recognition to aid UGV navigation helps human operators to make the critical decisions and even to take control of critical events (National Research Council 2002, vii).

2. Tactical Changes

There has been a shift in combat emphasis from head-on conventional war to low unit unconventional tactics; also there has been a shift in operating terrain from vegetated to urbanized theaters. Modern warfare and peacekeeping missions are now much more likely to take place in a built-up city. The UGV enables a force to handle a mission with fewer personnel, is capable of a more rapid deployment, and is easier to integrate into future digital battlefields. There has been little progress made in navigation through urban areas without human intervention (National Research Council 2005, 135). Even if there are good maps and a GPS receiver, urban navigation is always a challenge. There is always a limitation to the accuracy for both GPS and maps. For example, the GPS map developers may have to illustrate landmark features too for narrow roads and alleys so not clearly visible on maps (Glenn and Kingston 2005, 49).

During a tactical operation in an urban environment, navigation of UGV using GPS is challenging. There is a need to have redundancy to support or back up the GPS. Equipment such as static sensors, camera sensors, laser scanners can be considered. (Bonnifait et al. 2008, 84). The camera sensor is a popular choice as it is capable of collecting tremendous amounts of information for the unmanned system. An image capture provides many features that aid the UGV in understanding the environment and planning its next course. This thesis studies the applicability of using camera sensor harnesses with deep learning based methods to make improvements in accuracy and to make quick decisions in real time applications for UGV navigation.

B. USING ELECTRO-OPTICAL/INFRARED SENSORS

An electro-optical (EO) sensor (see Figure 2) operates like a camera that can be used to detect, recognize and identify objects such as human, vehicles, building and others in a long distance expecially at poor illumination environment (Keller 2013). The most commonly deployed EO sensors are image intensifier and thermal imager. Both EO sensors are able to operate in the day and night condition especially total darkness condition. There are two types of image intensifiers, the passive image intensifier and the active image intensifier (Kruegle 2011, 472). The passive image intensifier makes use of the natural illumination such as sun, moon, stars, and others to identify an object throught the object’s reflection (Kruegle 2011, 472). Whereas the active image intensifier emits invisible infra red enegy on the objects to identify the objects (Kruegle 2011, 472). The thermal imager identify an object through its emited radiation (Kruegle 2011, 469).

Figure 2. Picture of EO Camera for UGVs. Source: FLIR (2017).

The EO sensors provide situation awareness and useful information to infer the operation condition and understand the environment (Winnefeld and Kendall 2011, 47). This information can be fused with other static sensors to support the UGVs and operators on decision-making, identification, and tracking of threats (Winnefeld and Kendall 2011, 49). Figure 3 shows a picture of UGV mounted with an EO sensor.

Figure 3. Picture of UGV with EO Camera. Source: Studies Board and National Research Council (2005).

C. DEEP-LEARNING PARADIGM

Artificial intelligence, or AI, is the development of intelligent and user friendly applications to help humans solve problems efficiently. Artificial intelligence has evolved into many new areas of technology that can be integrated together to form a larger-scale system. It consists of many capabilities such as machine learning, speech recognition, optical character processing, and others (Norvig et al. 1995).

(1) Machine Learning

Machine learning (see Figure 4) is a subfield of AI that involves several scientific domains including mathematics, computer science, physics, and biology (Schapire 2008, 1). It can automatically find natural patterns, learn and make predictions from the collected information stored in the database (Murphy 2014, 1). The learning algorithms adaptively improve the output results with computational methods to make accurate predictions. Thus, it produces an output to help humans to make better decisions (Murphy 2014, 1).

The learning algorithm in machine learning makes use of manual feature extraction such as edges or corners and historical information to label images or recognize voices. Machine learning is fundamentally related to data analysis and statistics; therefore, the accuracy of the results depends on the quality of information provided and sample size of the information (Mohri et al. 2014, 1).

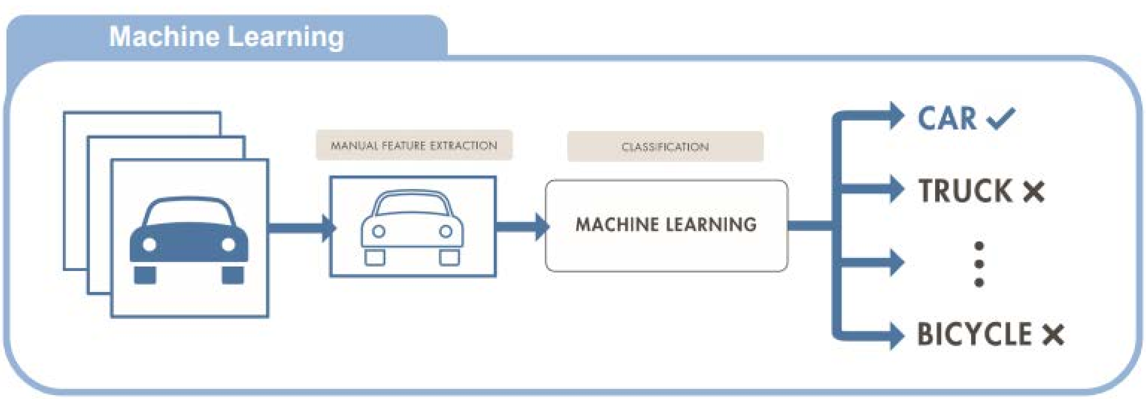

Figure 4. Machine Learning Workflow. Source: John (2017, 5).

The user is able to utilize the collected information to redefine the parameters of a system or model to optimize the solution. Besides that, it can use historical data to make predictions. Some of the applications for which machine learning can be deployed include:

- text or document classification
- natural language processing
- speech recognition
- optical character recognition
- computer vision such as image recognition and face detection
- autonomous vehicle navigation

(2) Deep-Learning

There is a smaller subcategory of machine learning called deep-learning. Deep-learning (see Figure 5) automatically extracts image features from large repository of training image dataset (Vinciarelli and Camastra 2015). The repository of training image dataset enables the machines to learn to classify the test images automatically. In short, the deep learning software would learn to recognize images that contain an object such as a car, without knowing what a car looks like (Marr 2016).

Deep-learning skips the manual step of extracting features from the images to classify the data, as opposed to most traditional machine learning algorithms, which require intense time and effort. However, deep-learning requires a few thousand images to get reliable results. Besides that, it requires a high performance GPU so that the system requires less time to analyze.

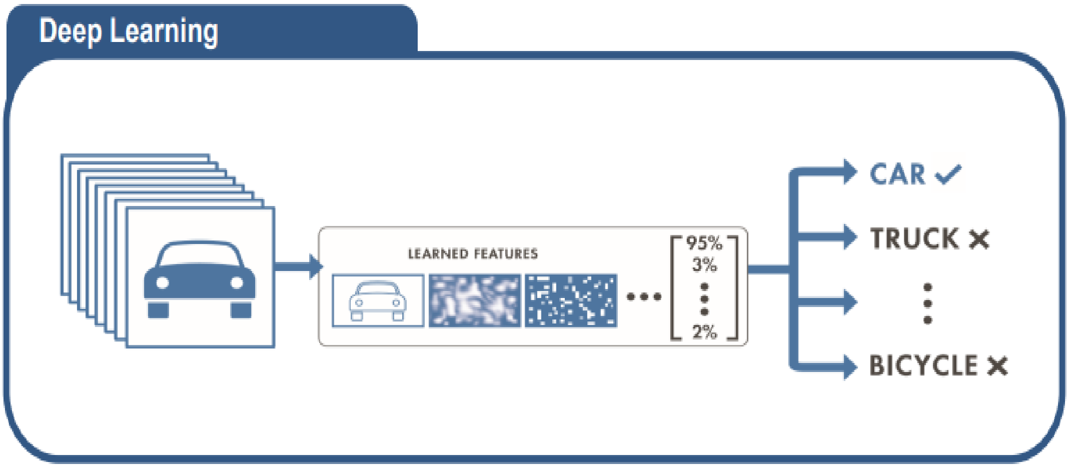

Figure 5. Deep-Learning Workflow. Source: John (2017, 5).

Deep-learning technology mainly makes use of a neural network architecture. The term "deep" refers to the hidden layers in the neural network. In the traditional neural network, it contains only two to three hidden layers, while the recent deep networks have as many as 150 (Mathworks 2017a). It is suited for image recognition to improve human problems such as optical character recognition, facial recognition, and many advanced driver assistance technologies such as autonomous driving, autonomous parking, and others. Table 1 shows the differences between machine learning and deep-learning.

Table 1. Differences between Machine Learning and Deep-Learning Technology

	Machine Learning	**Deep-Learning Technology**
Training database	Small	Large
Features	Yes	No
No. of Classifiers available	Many	No
Training time	Short	Few
Accuracy	Accurate	Highly Accurate

D. PROBLEM STATEMENT

Unmanned systems are still susceptible to technological limitations such as unstable communication and limited operating range. In an urban environment, there is high chance of GPS signal interruption due to physical structures such as concrete and steel walls, climate, and other factors including electromagnetic compatibility (EMC), electromagnetic interference (EMI) or media hub that may affect the UGV operations. Therefore, it is a challenge to receive consistent GPS and communication signals in the urban terrain (Glenn and Kingston 2005, 91). Over-reliance on communication technology only, including satellite communications that serve the GPS, will have significant operational risk (Blackburn et al. 2001, 92). Therefore, there is a need to explore technology and operational solutions that capitalize upon local autonomy and reduce communication requirements. The objective of this thesis is to explore applicability of deep-learning technology for UGV navigation in a GPS-degraded environment.

E. RESEARCH QUESTIONS

This thesis addresses the following research questions:

1. Can deep-learning technique that makes use of the preliminary trained neural networks be reliable in detecting and recognizing static and dynamic objects?

2. Can cognitive object recognition/classification aid in navigation of UGV?

3. Can cognitive object recognition/classification aid operators to make better decisions over the control of UGV navigation?

F. ORGANIZATION OF THESIS

To address the problem formulated in Section D, this thesis is structured in five chapters. After this background chapter, Chapter II highlights the software design and implementation, followed by Chapter III presenting on the system design. Chapter IV discusses the results of the experiment and challenges that were conducted using a Pioneer UGV. Chapter V concludes the work and provides some recommendations.

II. SOFTWARE DESIGN

This chapter explores and analyzes the development of the following software algorithm for graphical user interface (GUI), deep learning technique and cascade object detector to aid the operators on UGV navigation. These three software algorithms shall be further described in the following sections.

A. GRAPHICAL USER INTERFACE

GUI development environment (GUIDE) is a feature in the MATLAB that allows the software developer to design and develop a user-friendly GUI for the operators. GUIDE provides various interactive buttons and controls that the operators are able to start/stop streaming of live video feed, capture, display and save images to the storage. The GUI provides the operators situation awareness by showing the field of view of the cameras. In addition, the captured image will undergo image recognition using deep learning technique and display the captured image with object name to aid the operators in making decisions on the command and control of the UGV (see Figure 6).

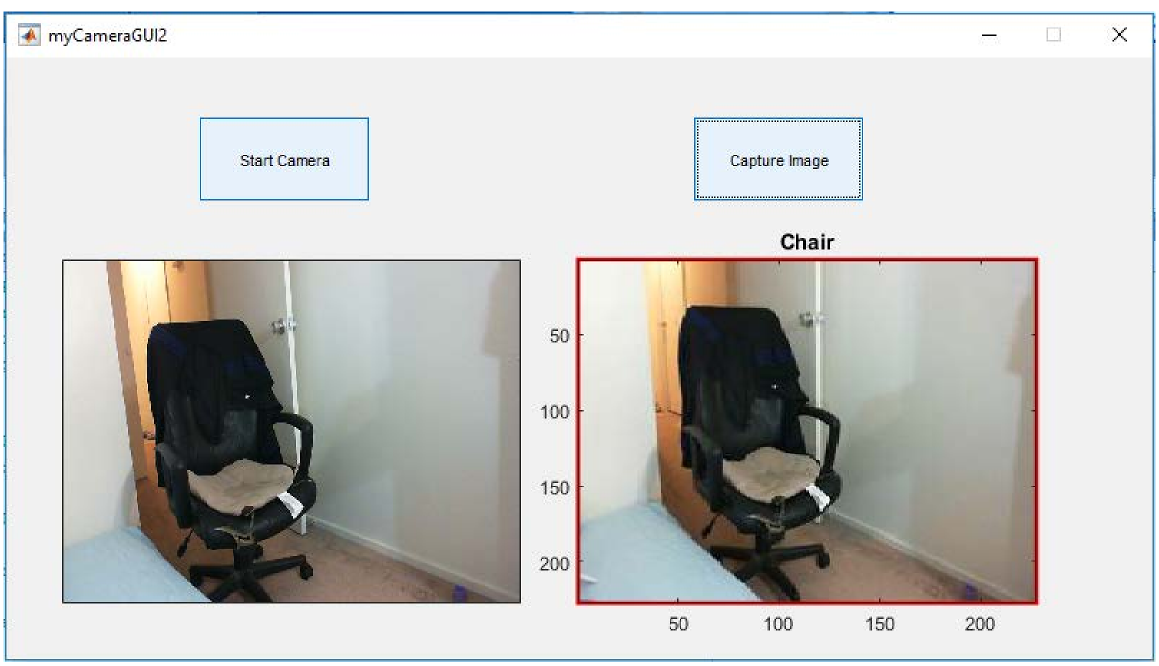

Figure 6. Graphical User Interface (GUI)

B. DEEP-LEARNING TECHNIQUE

The convolutional neural network (CNN) is one of the machine learning algorithms that can be found in deep-learning. (Mathworks 2017a). It automatically extracts features from a large collection of images for image classification and object detection (Mathworks 2017a). At the current state of research, there are three techniques that can be successfully deployed for CNN on image classification. Table 2 describes the three different types of CNN techniques.

Table 2. CNN Techniques. Source: Shin et al. (2016).

S/N	CNN Techniques	Analysis
1	Training the CNN from scratch	To create a new convolution network, it requires a large dataset which is challenging, time consuming and ineffective to build.
2	Using existing pretrained features	Using off-the-shelf pretrained features to perform image classification using CNN.
3	Transfer learning approach	Transfer learning makes use of existing pretrained features to transfer of the knowledge or learned features to solve a new problem. Developed with small training data, it is more practical to use an existing pretrained model to improve the image classification accuracy. In addition, the training can be completed faster as it only take the last few layers from pretrained network and fine-tuned to learn the features of the new image collection.

There are many off-the-shelf pretrained networks available such as VGG-16, VGG-19, GoogleNet, and Alexnet. Alexnet is one of the most popular pretrained networks which has proven to obtain significantly good results over the video analytics methods especially in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 (Krizhevsky et al. 2012). The good results has garner lots of interest in deep-learning technology, thus it has been selected to fine-tune the system in this research. The Alexnet is one of the most studied CNN which comprises of feature learning and classification (Redmon and Farhadi 2016). Figure 7 depicts the workflow of input test image passing through the convolution layers, pooling layers, and fully-connected layers (John 2017, 12). It has 1.2 million of images with resolution of 256 x 256 pixels in the dataset and, up to 1000 image categories available for classification of objects (Shin et al. 2016). Figure 8 reflects the workflow on classification of test image using transfer learning approach.

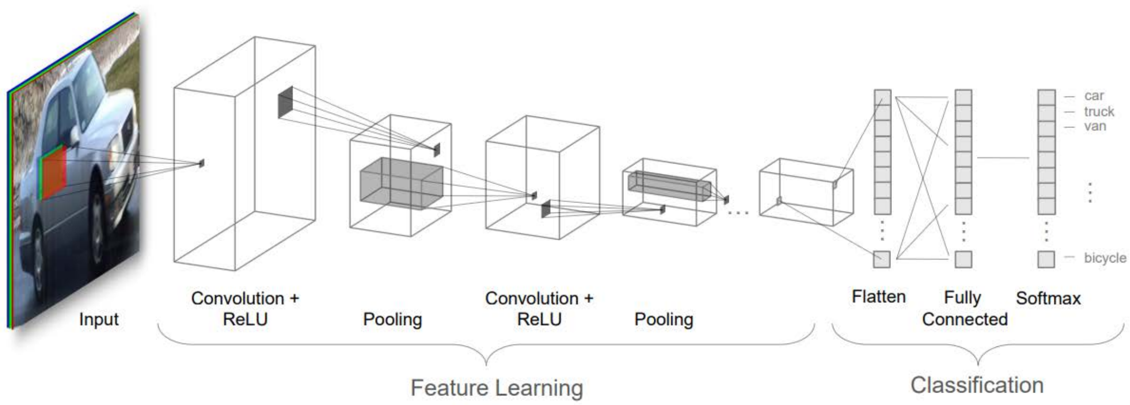

Figure 7. Convolution Neural Networks Workflow. Source: John (2017, 12).

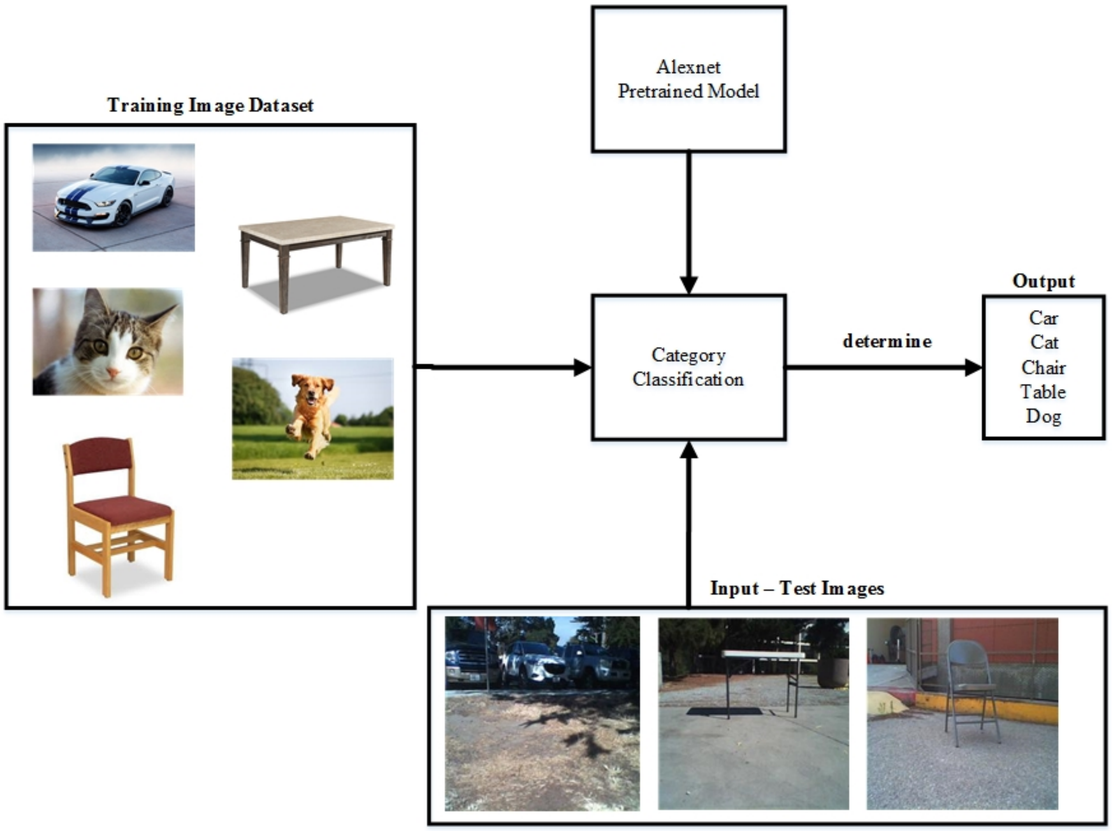

Figure 8. Workflow on Classification of Test Image using Transfer Learning Approach

Figure 9 illustrates the Matlab command on downloading the pretrained network, Alexnet, and specifying the folder that store the pretrained network.

```
% Location of pre-trained "AlexNet"
cnnURL = 'http://www.vlfeat.org/matconvnet/models/beta16/imagenet-caffe-alex.mat';
% Specify folder for storing CNN model
cnnFolder = 'D:\NPS\Thesis\Computer vision'
cnnMatFile = 'imagenet-caffe-alex.mat';
cnnFullMatFile = fullfile(cnnFolder, cnnMatFile);

% Check that the code is only downloaded once
if ~exist(cnnFullMatFile, 'file')
    disp('Downloading pre-trained CNN model...');
    websave(cnnFullMatFile, cnnURL);
end
```

Figure 9. Matlab Command on Downloading of Pretrained Network, Alexnet. Source: Mathworks (2017b).

Figure 10 shows the Matlab command to illustrate the architecture of the CNN. The Matlab output displays the number of layers in the CNN (Mathworks 2017b). The sequence of layers is align with the earlier discussion (see Figure 7) on the workflow of input test image passing through the convolution layers, pooling layers, and fully-connected layers (Mathworks 2017b).

```
%% |convnet.Layers| defines the architecture of the CNN

convnet.Layers
```

```
ans =

  23x1 Layer array with layers:

     1   'input'               Image Input                   227x227x3 images with 'zerocenter' normalization
     2   'conv1'               Convolution                   96 11x11x3 convolutions with stride [4  4] and padding [0  0]
     3   'relu1'               ReLU                          ReLU
     4   'norm1'               Cross Channel Normalization   cross channel normalization with 5 channels per element
     5   'pool1'               Max Pooling                   3x3 max pooling with stride [2  2] and padding [0  0]
     6   'conv2'               Convolution                   256 5x5x48 convolutions with stride [1  1] and padding [2  2]
     7   'relu2'               ReLU                          ReLU
     8   'norm2'               Cross Channel Normalization   cross channel normalization with 5 channels per element
     9   'pool2'               Max Pooling                   3x3 max pooling with stride [2  2] and padding [0  0]
    10   'conv3'               Convolution                   384 3x3x256 convolutions with stride [1  1] and padding [1  1]
    11   'relu3'               ReLU                          ReLU
    12   'conv4'               Convolution                   384 3x3x192 convolutions with stride [1  1] and padding [1  1]
    13   'relu4'               ReLU                          ReLU
    14   'conv5'               Convolution                   256 3x3x192 convolutions with stride [1  1] and padding [1  1]
    15   'relu5'               ReLU                          ReLU
    16   'pool5'               Max Pooling                   3x3 max pooling with stride [2  2] and padding [0  0]
    17   'fc6'                 Fully Connected               4096 fully connected layer
    18   'relu6'               ReLU                          ReLU
    19   'fc7'                 Fully Connected               4096 fully connected layer
    20   'relu7'               ReLU                          ReLU
    21   'fc8'                 Fully Connected               1000 fully connected layer
    22   'prob'                Softmax                       softmax
    23   'classificationLayer' Classification Output         cross-entropy with 'n01440764', 'n01443537', and 998 other classes
```

Figure 10. Matlab Command on the Architecture of CNN

This thesis focuses on three different types of training images datasets to compare the system accuracy, as shown in Table 3.

Table 3. Three Different Types of Image Datasets

Types of training datasets	Description
Dataset with 20 training images	Collection of 20 training images each of predefined categories.
Dataset with original 20 plus new 20 training images	Collection of original 20 plus 20 new training images each of pre-defined categories.
Dataset with 39 training images from above plus 1 image of the actual scene	Collection of 39 training images from the above plus 1 image taken from an actual scene with the object of interest.

The training images dataset are specifically targeting on five types of categories (see Figure 11). The five types of categories are Chair, Table, Car, Cat, and Dog. Figure 12 displays the number of training images in each of the five categories.

```
%% Set up image data
dataFolder = 'D:\NPS\Thesis\Computer vision\Images'
categories = {'Cat', 'Dog', 'Car', 'Table', 'Chair'};
imds = imageDatastore(fullfile(dataFolder, categories), 'LabelSource', 'foldernames');
tbl = countEachLabel(imds)
```

Figure 11. Matlab Command on Specifying Five Types of Categories

```
tbl =

    Label    Count
    _____    _____

    Car      40
    Cat      40
    Chair    40
    Dog      40
    Table    40
```

Figure 12. Number of Training Images in Each of the Five Categories

The CNN algorithm can only process red, green blue (RGB) images with dimension of width at 227 pixels and height at 227 pixels (Mathworks 2017b). Figure 13 details the extraction of the training features such as edges and blobs from the training images to train the software algorithm.

```
%% Extract training features using pretrained CNN

% Get the network weights for the second convolutional layer
w1 = convnet.Layers(2).Weights;

% Scale and resize the weights for visualization
w1 = mat2gray(w1);
w1 = imresize(w1,5);

% Display a montage of network weights.
figure
montage(w1)
title('First convolutional layer weights')

featureLayer = 'fc7';
trainingFeatures = activations(convnet, trainingSet, featureLayer, ...
    'MiniBatchSize', 32, 'OutputAs', 'columns');

% Get training labels from the trainingSet
trainingLabels = trainingSet.Labels;

classifier = fitcecoc(trainingFeatures, trainingLabels, ...
    'Learners', 'Linear', 'Coding', 'onevsall', 'ObservationsIn', 'columns');
```

Figure 13. Matlab Command on Extraction of Training Features

In Figure 14, the algorithm extracts the features from the test image and allows it to make a prediction on classifying the test image (Mathworks 2017b). The accuracy is the measure on classifying the test image correctly.

```
% Extract test features using the CNN
testFeatures = activations(convnet, testSet, featureLayer, 'MiniBatchSize',32);

% Pass CNN image features to trained classifier
predictedLabels = predict(classifier, testFeatures);

% Get the known labels
testLabels = testSet.Labels;

% Tabulate the results using a confusion matrix.
confMat = confusionmat(testLabels, predictedLabels);

% Convert confusion matrix into percentage form
confMat = bsxfun(@rdivide,confMat,sum(confMat,2))


img = readAndPreprocessImage(newImage);

imageFeatures = activations(convnet, img, featureLayer);
label = predict(classifier, imageFeatures)
image(img)
title(char(label));

accuracy = sum(predictedLabels==testLabels)/numel(predictedLabels)
```

Figure 14. Matlab Command on Extraction of Test Features

C. CASCADE-OBJECT DETECTOR

The developed deep learning algorithm is unable of provide accurate bounding box on the recognize objects; therefore, cascade-object detector was explored. It is another method of learning-based solution which makes use of large collection of both positive and negative images to train system on detection and recognized object with bounding box (Mathworks 2017c). The training images will be processed under cascade classifier to label the object of interest (Mathworks 2017c). There are multiple training stages in the cascade classifier to reduce the false negative rate on incorrectly labeling the objects (Mathworks 2017c). Figure 15 describes the workflow of cascade-object detector. Figure 16 is an example of a recognized object with bounding box.

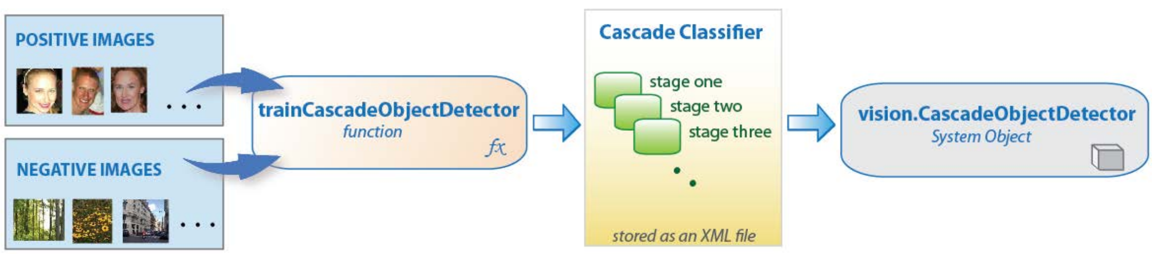

Figure 15. Cascade-Object Detector Workflow. Source: Mathworks (2017c).

Figure 16. Recognized Object with Bounding Box

THIS PAGE INTENTIONALLY LEFT BLANK

III. SYSTEM DESIGN

This chapter details the overall architecture, hardware and software, to be carried out for this research.

A. HARDWARE

1. Pioneer UGV

The 12 kg Pioneer UGV as shown in Figure 17 is a robot with two-wheel and two-motor differential drive. It is best suited for research and experiment for an indoor laboratory. The 0.5 m width robots with 0.2 m diameter drive wheels is used for research due to its versatility, reliability and durability. It has an endurance of up to four hours with a forward speed of 0.7m/s. It is capable of carrying up to 30 kg payload if it is maneuvering at slow speed on a flat terrain. The baseline UGV is equipped with a computer running on the Linux Ubuntu 14.04 operating system with the robot operating system (ROS) packages that generate the command for maneuvering the UGV.

The Microstrain sensor (P/No: 3DM-GX3 -45) was installed on all the Pioneer UGVs that have a GPS-aided Inertial Navigation System (IMU/GPS). It combines the MEMs inertia sensors with the embedded GPS receiver and the extended Kalman Filter algorithm to generate optimal position estimated for the robots.

Figure 17. Pioneer UGV

2. Camera Sensor

A camera sensor consists of a lens, an image sensor, and other supporting electronic components. The camera lens provides clear images of the object/scene for the camera sensor. The size of the camera lens will determine the field of view of the video feed. The image sensor receives light through the camera lens and convert the object/scene information into an image. The following sections provide a list of different types of camera sensors:

a. Web Camera

A web camera (see Figure 18) is a digital video camera that streams real-time high definition (HD) resolution (up to 1920 x 1080 pixels) video via universal serial bus (USB) connection to a computer. The web camera is commonly used for video calling over an Internet connection, even though it is capable to be used for security system purposes. It is a simple and cheap device that can easily be set up by any consumer. However, it has limited camera features such as adjustment of video resolution, shutter speed and sensitivity noise ratio. In addition, there is also distance limitation between the web camera and the computer due to the USB cable.

Figure 18. Sample of Web Camera. Source: Logitech (2017).

b. Network Camera

A network camera also known as an IP camera (see Figure 19), is also a digital video camera, but it has its own IP address like a network device. It can be connected to a network system by physical network cable or by a wireless network connection. It is unlike the web camera which can only be connected to a computer by USB connection. The web camera can only operate with installed software on a computer, whereas the network camera operates like a web server. Operators can access the web browser via Hypertext Transfer Protocol (HTTP). The web browser allows the system administrator to adjust the video resolution from video graphics array (VGA) (640 x 480 pixels) to HD (1920 x 1080 pixels) video resolution, shutter speed and sensitive noise ratio. The adjustment depends on the availability of network throughput for streaming high video resolution.

Figure 19. Network Camera. Source: D-Link (2017).

Although the network camera is more complex and requires basic technical knowledge, it offers more features and better image quality. The cost of network cameras will be higher than the web camera.

In this research, Ai-Ball is selected for being an exceptionally small wireless network camera that can be mounted on UGV with limited space (See Figure 20). The design of the camera can easily blend into the UGV for discreet surveillance. The camera is capable of streaming live video feed wirelessly and capture image for object detection and recognition. Figure 20 shows the picture of an Ai-Ball mounted on the UGV.

Figure 20. Camera Sensor Ai-Ball

The key performance parameters (KPPs) specification of the Ai-Ball camera are listed in Table 4.

Table 4. Key Performance Parameters (KPPs) Source: Ai-Ball (2017).

Ai-Ball	**Specification**
Video Resolution	VGA 640x480, QVGA 320x240, QQVGA 160x120, up to 30fp
View Angle	60 degree
Wireless Interface	IEEE 802.11b/g 2.4GHz ISM Band
Wireless Security	WEP 64/128, WPA, WPA2
Wireless Range	• Infrastructure: 20m (Typical) • Adhoc: 7.5m (Typical)
Dimension	30mm(Diameter) x 35mm(L)
Weight	100g
Power Supply	• Battery operated • Voltage: 3.0V • Power: 750mAH (CR2) • Current consumption: 320mA (typical); 350mA (maximum)

B. SOFTWARE

Figure 21 depicts the software component and software functional flow diagram for the UGV. Ubuntu 14.04 was the operating system software for the remote workstation, and it connects wirelessly to the UGV via a wireless router. Matlab was used to develop and run the source code for UGV control and other functionalities. The Matlab communicates with ROS to command and control the UGV. The details of the software component will be discussed further in the following sub-section.

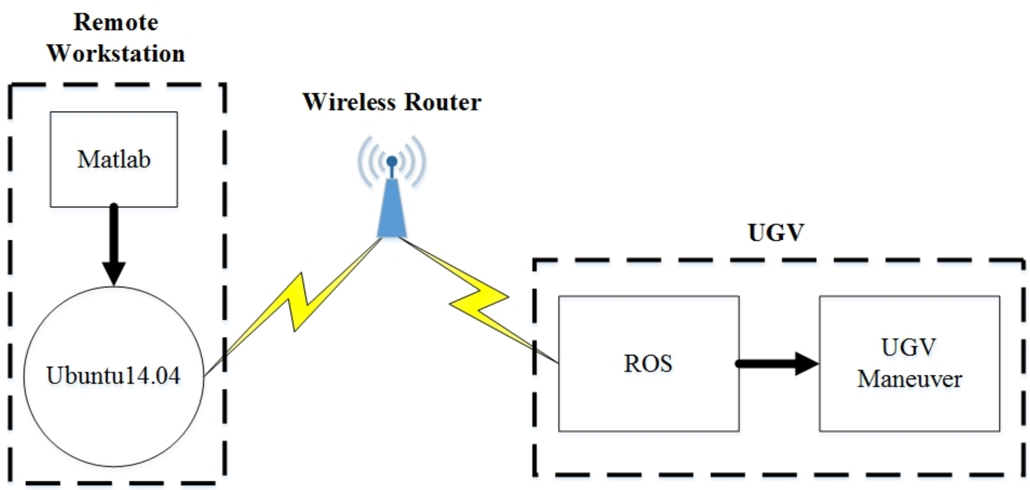

Figure 21. Software Functional Flow Diagram

1. MATLAB 2016a

The Matlab is the platform utilized to develop the source code for the robotic control and deep-learning algorithm. In addition, the Matlab has Robotics System Toolbox that provides the support to interface with ROS and ROS interface.

2. Ubuntu 14.04

One of many distributions of Linux for personal computers and other Internet of Things (IOT) devices. It is a simplified software distribution that is well integrated with ROS. In this research, the Ubuntu is the operating software that interface with both MATLAB and ROS. MATLAB shall communicates with the ROS network wirelessly using Ubuntu wireless interface.

3. Robot Operating System Indigo Igloo (ROS)

Robot Operating System (ROS) is a software development platform for developers to create source code for robot command and control. The software was originally developed at Stanford AI Lab and is currently maintained by Willow Garage. It offers software libraries and tools to help a software developer build a software application for robot. It acts as a middleware that provide inter-process communication by enabling programs (Nodes) to communicate.

IV. EXPERIMENTAL RESULTS

Three test scenarios were conducted in this thesis research to verify and validate the image classification accuracy through the developed deep-learning algorithm. The three test scenarios are identification of a single object which consists of both indoor and outdoor environment tests, identification of multiple homogeneous objects, and identification of multiple heterogeneous objects. In each test scenario, the system shall perform 10 test cycle runs for data collection. The purpose of 10 test cycle runs was to test, verify and validate that the system can consistently provide accurate results. This section starts from showing and discussing the results of each of the aforementioned test scenarios and concludes with a discussion at some of the challenges faced during the experiment tests.

A. IDENTIFICATION OF A SINGLE OBJECT

There were two tests conducted on identification of a single object. The first test was a chair along a walkway at an indoor environment. While the second test was a table at an outdoor environment. The Pioneer UGV was navigated to the target of interest to perform image classification test. The sample pictures of an identified single object in indoor environment are shown in Figure 22.

1. Indoor Environment Test on Single Object

Figure 22. Indoor Environment Test on Single Object—Chair

Table 5 summarizes the results of the indoor environment test conducted on a single object. It shows that the percentage of correct identification of the system improved with the increase in number of training images. The system was able to achieve 100% success in identifying the object correctly when one of the training images was replaced by an image of the actual scene, the chair. The results from Table 6 to Table 8 detail the three test cases such as dataset with 20 training images, dataset with original 20 plus new 20 training images and dataset with 39 training images from above plus 1 image of the actual scene that were conducted. The tables consist of confidence level, results and pass/fail. Confidence level is the calculation on how confidence the deep-learning algorithm recognizing the object correctly. The results are classification of image by the deep-learning software algorithm and finally the pass/fail is to show if the software has recognize the object correctly. When the object is recognize correctly, the table will be updated as pass. While fail will be given to image been recognized wrongly.

Table 5. Summary Table of Percentage of Correct Identification on Indoor Environment Test on a Single Object

Test Case	**Percentage of Correct Identifications**
Dataset with 20 training images	30%
Dataset with original 20 plus new 20 training images	80%
Dataset with 39 training images from above plus 1 image of the actual scene	100%

Table 6. Results of Indoor Environment Test on a Single Object with 20 Training Images in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.90	0.93	0.94	0.86	0.87	0.91	0.93	0.96	0.89	0.93
Results	Dog	Table	Table	Chair	Table	Chair	Table	Chair	Table	Table
Pass / Fail	Fail	Fail	Fail	Pass	Fail	Pass	Fail	Pass	Fail	Fail

Table 7. Results of Indoor Environment Test on a Single Object with Original 20 Plus New 20 Training Images in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.95	0.94	0.92	0.94	0.93	0.92	0.94	0.92	0.95	0.91
Results	Chair	Chair	Chair	Chair	Table	Car	Chair	Chair	Chair	Chair
Pass / Fail	Pass	Pass	Pass	Pass	Fail	Fail	Pass	Pass	Pass	Pass

Table 8. Results of Indoor Environment Test on a Single Object with 39 Training Images from the above Plus 1 Image of the Actual Scene in the Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.90	0.96	0.93	0.91	0.91	0.95	0.94	0.96	0.91	0.90
Results	Chair	Chair	Chair	Chair	Chair	Chair	Chair	Chair	Chair	Chair
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

2. Outdoor Environment Test on Single Object

The second test on identification of a single object was conducted at an outdoor environment and the target was a table. It is more challenging to conduct an outdoor test due to several factors such as lighting, moving background objects, shadows, and sun glare. The Pioneer UGV was navigated to the target of interest to conduct the image classification test. The sample pictures of an identified single object in outdoor environment are shown in Figure 23.

Figure 23. Outdoor Environment Test on Single Object—Table

Table 9 summarizes the results of the outdoor environment test conducted on a single object. The system was able to achieve 80% success in identifying the single object at outdoor environment correctly when one of the training images was replaced by an image of the actual scene, the table. Detail results of the three test cases such as dataset with 20 training images, dataset with original 20 plus new 20 training images and dataset with 39 training images from above plus 1 image of the actual scene that were conducted are shown from Table 10 to Table 12.

Table 9. Summary Table of Percentage of Correct Identification on Outdoor Environment Test on a Single Object

Test Case	**Percentage of Correct Identifications**
Dataset with 20 training images	70%
Dataset with original 20 plus new 20 training images	70%
Dataset with 39 training images from above plus 1 image of the actual scene image	80%

Table 10. Results of Outdoor Environment Test on a Single Object with 20 Training Images in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.93	0.86	0.91	0.89	0.90	0.86	0.91	0.87	0.90	0.90
Results	table	table	table	table	table	Chair	Chair	table	table	Chair
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Fail	Fail	Pass	Pass	Fail

Table 11. Results of Outdoor Environment Test on a Single Object with Original 20 Plus New 20 Training Images in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.96	0.92	0.89	0.93	0.93	0.88	0.94	0.87	0.94	0.89
Results	table	table	Chair	table	table	table	Chair	Chair	table	table
Pass / Fail	Pass	Pass	Fail	Pass	Pass	Pass	Fail	Fail	Pass	Pass

Table 12. Results of Outdoor Environment Test on a Single Object with 39 Training Images from above Plus 1 Image of the Actual Scene in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.91	0.91	0.91	0.92	0.91	0.90	0.89	0.84	0.91	0.90
Results	table	table	table	table	table	Chair	Chair	table	table	table
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Fail	Fail	Pass	Pass	Pass

B. IDENTIFICATION OF MULTIPLE HOMOGENEOUS OBJECTS

Identification of multiple homogeneous objects were conducted at an indoor environment and the target was two chairs within the field of view in the on-board camera. The Pioneer UGV was navigated to the target of interest to perform image classification test. The sample pictures of identified multiple homogeneous objects are shown in Figure 24.

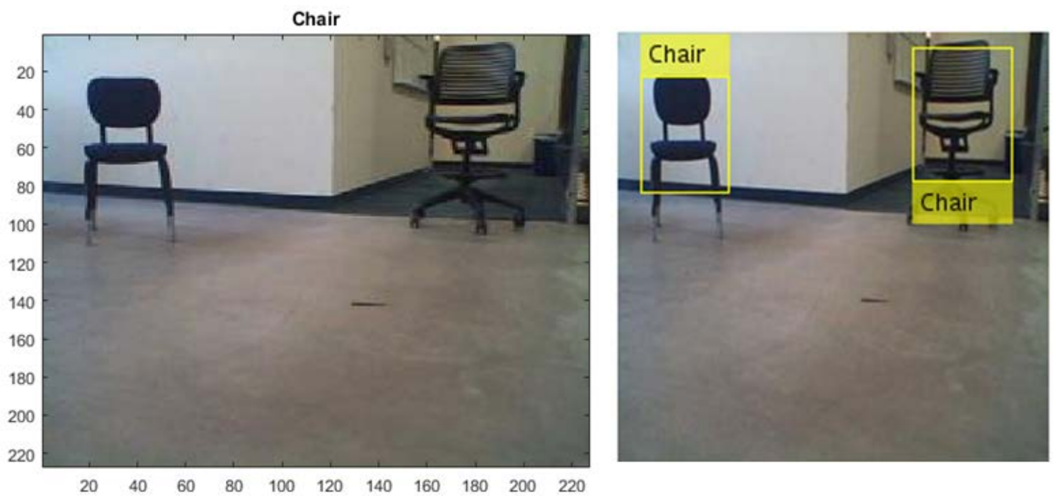

Figure 24. Identification of Multiple Homogeneous Objects

Table 13 summarizes the results of test conducted on multiple homogeneous object. Comparing with the earlier test scenario, the system has obtained slightly better results in dataset with 20 training images. Although there is a slight reduction in the percentage of correct identification on the dataset with 40 training images, the system was able to achieve 100% success in identifying the multiple homogeneous objects correctly when one of the training images was replaced by an image of the actual scene, the two chairs. Detail results of the three test cases such as dataset with 20 training images, dataset with original 20 plus new 20 training images and dataset with 39 training images from above plus 1 image of the actual scene that were conducted are shown from Table 14 to Table 16.

Table 13. Summary Table of Percentage of Correct Identification on Multiple Homogeneous Objects

Test Case	**Percentage of Correct Identifications**
Dataset with 20 training images	50%
Dataset with original 20 plus new 20 training images	60%
Dataset with 39 training images from above plus 1 image of the actual scene image	100%

Table 14. Test Results of Multiple Homogeneous Objects with 20 Training Images in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.89	0.89	0.84	0.94	0.93	0.86	0.89	0.93	0.91	0.91
Results	Chair	Table	Chair	Chair	Table	Table	Table	Table	Chair	Chair
Pass / Fail	Pass	Fail	Pass	Pass	Fail	Fail	Fail	Fail	Pass	Pass

Table 15. Test Results on Multiple Homogeneous Objects with Original 20 Plus New 20 Training Images in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.94	0.94	0.92	0.88	0.89	0.94	0.91	0.97	0.94	0.92
Results	Chair	Table	Chair	Chair	Table	Chair	Table	Table	Chair	Chair
Pass / Fail	Pass	Fail	Pass	Pass	Fail	Pass	Fail	Fail	Pass	Pass

Table 16. Test Results on Multiple Homogeneous Objects with 39 Training Images from above Plus 1 Image from the Actual Scene in Dataset

	Test Run									
	1	2	3	4	5	6	7	8	9	10
Confidence level	0.93	0.94	0.94	0.89	0.89	0.89	0.96	0.91	0.97	0.92
Results	Chair	Chair	Chair	Chair	Chair	Chair	Chair	Chair	Chair	Chair
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

C. IDENTIFICATION OF MULTIPLE HETEROGENEOUS OBJECTS

Identification of multiple heterogeneous objects was conducted outdoors and the Pioneer UGV was expected to navigate to the three selected targets (Table, Car, and Chair) at pre-defined location to perform the test. The setup of predefined targets' location and pioneer UGV navigation path are shown in Figure 25. And the sample pictures of identified multiple heterogeneous objects at predefined positions are shown Figure 26.

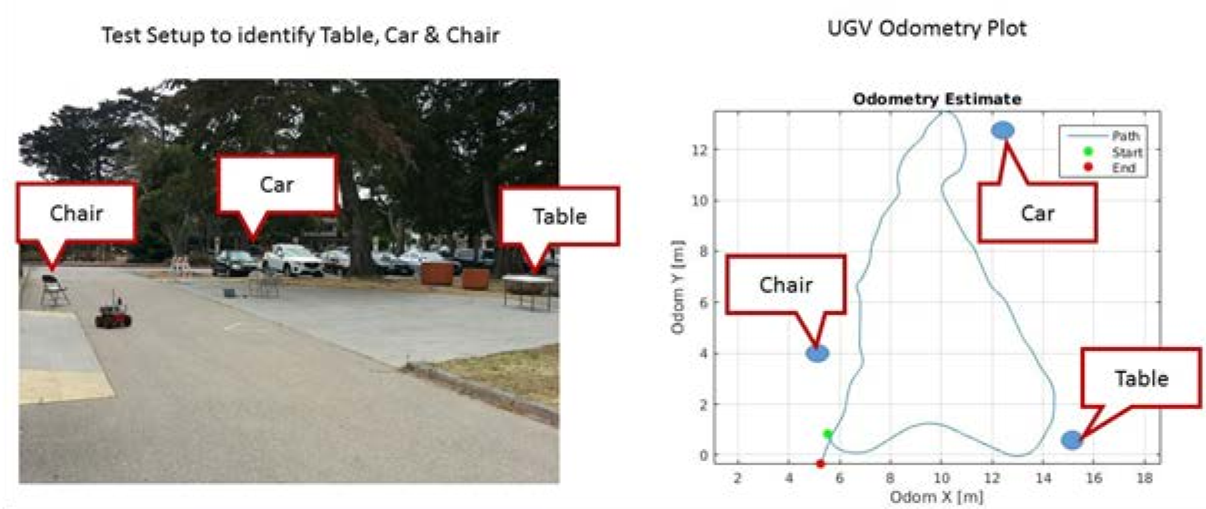

Figure 25. Setup of Predefined Targets' Location and Pioneer UGV Navigation Route

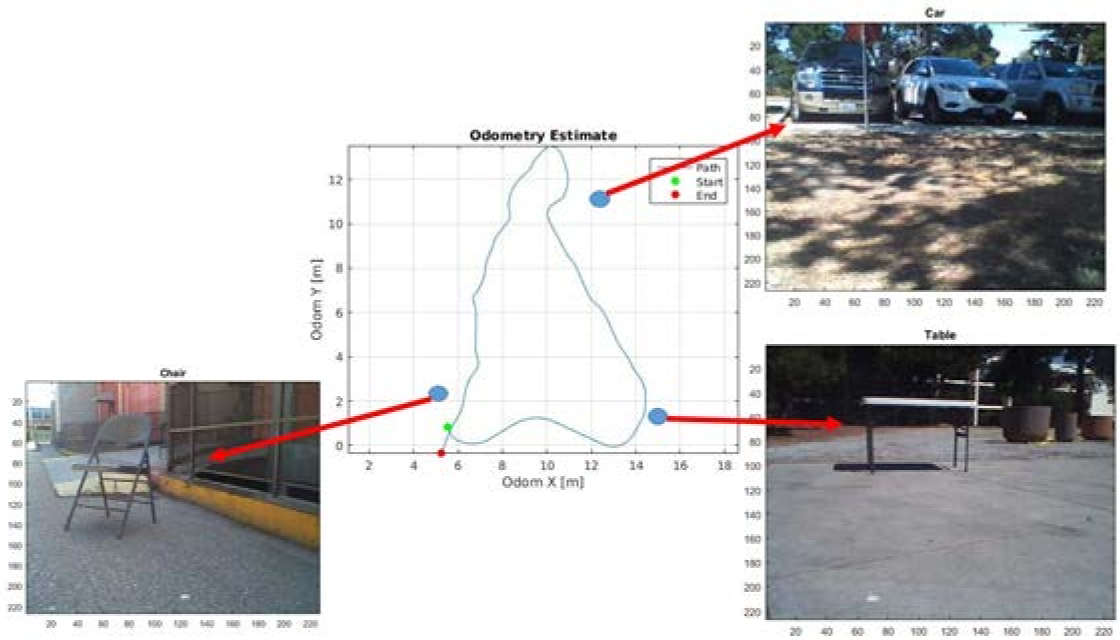

Figure 26. Recognized Images of Targets at Predefined Position

Table 17 summarizes the results of test conducted on multiple heterogeneous object. The system has successfully demonstrated its capability in recognizing the pre-defined targets. The system was able to achieve 93% in identifying the multiple heterogeneous objects correctly on test case with 39 training images from earlier dataset plus 1 image of the actual scene. Detail results of the three test cases such as dataset with 20 training images, dataset with original 20 plus new 20 training images and dataset with 39 training images from above plus 1 image of the actual scene that were conducted are shown from Table 18 to Table 20.

Table 17. Summary Table of Percentage of Correct Identification on Multiple Heterogeneous Objects

Test Case	**Percentage of Correct Identifications**
Dataset with 20 training images	70%
Dataset with original 20 plus new 20 training images	87%
Dataset with 39 training images from above plus 1 image of the actual scene	93%

Table 18. Test Results on Multiple Heterogeneous Objects with 20 Training Images in Dataset

	Test Run																													
	1			2			3			4			5			6			7			8			9			10		
Confidence level	0.91	0.93	0.90	0.93	0.91	0.87	0.91	0.89	0.89	0.87	0.91	0.84	0.86	0.90	0.93	0.90	0.90	0.90	0.87	0.89	0.86	0.93	0.90	0.87	0.93	0.90	0.90	0.91	0.89	0.86
Results	Table	Car	Table	Chair	Car	Table	Table	Dog	Chair	Table	Car	Table	Table	Car	Chair	Table	Car	Chair	Chair	Table	Table	Table	Dog	Chair	Table	Car	Chair	Table	Car	Chair
Pass / Fail	Pass	Pass	Fail	Fail	Pass	Fail	Pass	Fail	Pass	Pass	Pass	Fail	Pass	Pass	Pass	Pass	Pass	Pass	Fail	Fail	Fail	Pass	Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 19. Test Results on Multiple Heterogeneous Objects with Original 20 Plus New 20 Training Images in Dataset

	Test Run																													
	1			2			3			4			5			6			7			8			9			10		
Confidence level	0.95	0.94	0.89	0.94	0.92	0.92	0.96	0.92	0.94	0.91	0.91	0.94	0.94	0.93	0.91	0.91	0.94	0.91	0.93	0.96	0.92	0.89	0.93	0.92	0.93	0.88	0.94	0.87	0.94	0.89
Results	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Chair	Car	Table	Table	Car	Table	Table	Car	Chair
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Fail	Pass	Fail	Pass	Pass	Fail	Fail	Pass	Pass

Table 20. Test Results on Multiple Heterogeneous Objects with 39 Training Images from the above Plus 1 Image of the Actual Scene in Dataset

	Test Run														
	1			2			3			4			5		
Confidence level	0.90	0.94	0.95	0.89	0.91	0.91	0.91	0.92	0.91	0.94	0.91	0.90	0.92	0.95	0.91
Results	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Table	Car	Table
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Fail

	Test Run														
	6			7			8			9			10		
Confidence level	0.89	0.91	0.94	0.92	0.91	0.93	0.92	0.91	0.89	0.93	0.92	0.94	0.94	0.89	0.94
Results	Table	Car	Chair	Table	Car	Chair	Table	Car	Chair	Car	Car	Chair	Table	Car	Chair
Pass / Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Fail	Pass	Pass	Pass	Pass	Pass

D. CHALLENGES

During the course of experimental testing, there were some challenges faced which affect the outdoor test due to several factors such as lighting, moving background objects, shadows, and sun glare. The below figures show different types of challenges faced during the test process.

1. Direct Sunlight Glare

Typically, direct sunlight glare tends to happen right after sun rises and before sunset. The sunlight glare affects the field of view of the camera. As shown in Figure 27, the sunlight glare blocks out most parts of the chair which affected the system; it identified it as a table instead. To recognize the object correctly, the UGV needs to maneuver to a position to avoid the sunlight glare.

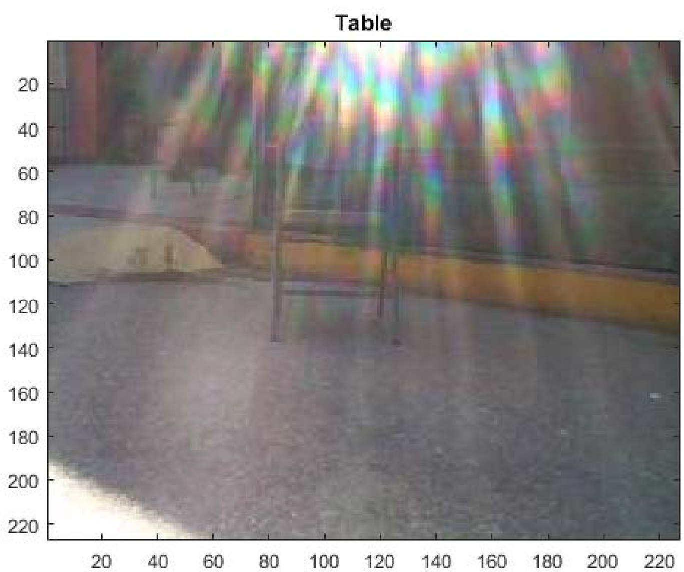

Figure 27. Sample of Picture Showing Direct Sunlight Glare Interfering with Detection Process

2. Shadow

The movement of the sun will cast a shadow of an object at different directions at different times of the day. A shadow of the object of interest may cause the system to interpret it as part of the object. Based on Figure 28, the table was recognized as a chair while the car was recognized as a dog.

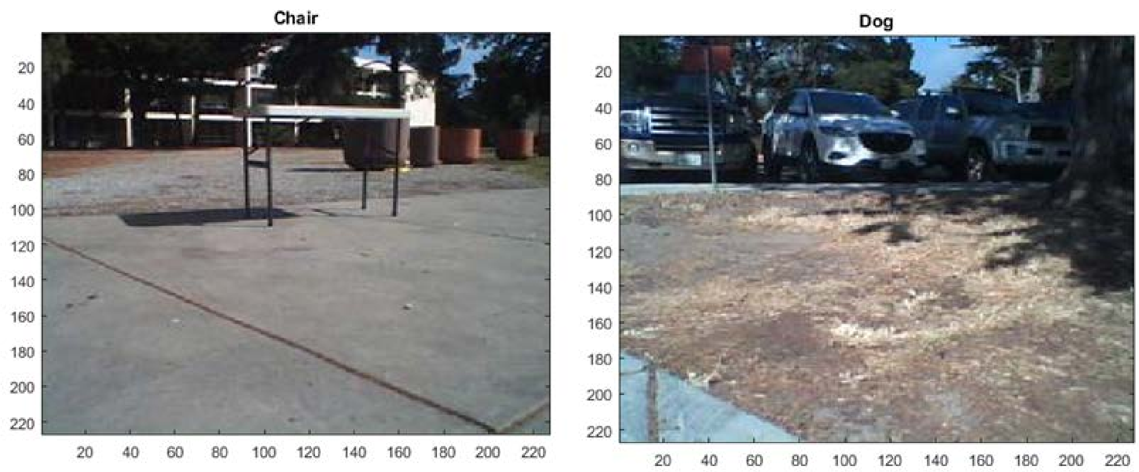

Figure 28. Samples of Pictures Showing Shadow Interfering with Detection Process

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

The aim of this thesis was to study the applicability of deep-learning technology for relative object-based navigation. The transfer learning approach technique was deployed with Alexnet as the pretrained network to improve the image recognition accuracy. Four types of different test scenarios were conducted to verify and validate that the system is able to detect and identify an object correctly. For example, based on the ten the results of all the test scenarios are summarized in Table 21.

Table 21. Summary of the Percentage of Correct Identification on the Four Types of Test Scenarios

	Percentage of Correct Identifications			
Test Case	**Chair**	**Two Chairs**	**Table**	**Table/Car/Chair**
Dataset with 20 training images	30%	50%	70%	70%
Dataset with original 20 plus new 20 training images	80%	60%	70%	87%
Dataset with 39 training images from above plus 1 image of the actual scene	100%	100%	80%	93%

Based on the results, the dataset with 39 training images from above plus 1 image of the actual scene obtained the overall best results. The good results could be attributed to having an actual image of the targets in the dataset. This thesis demonstrates that it addresses the earlier research questions on using deep-learning technology reliably detecting and recognizing static objects and it help the operators to make better decisions over the control of UGV navigation.

B. RECOMMENDATIONS

The list of recommendations for future work that can be carried out to expand on the work in this thesis include obstacle avoidance and target detection software.

Specifically, with the success in the earlier test scenario, it is recommended to further explore obstacle avoidance using deep-learning technique. This implementation would enable UGV to navigate autonomously without knocking into an object. The deep-learning technique shall assist the Pioneer UGV system to recognize the object and determine best route to avoid the obstacle.

Cascade-video detector was developed to assist the Pioneer UGV system on target identification. The techniques developed in this thesis happened to be sensitive to detecting the objects at different scales. Besides that, it requires huge number of positive and negative training images. The software only allows uploading of one XML file (of a specific target) in the Matlab software code. Therefore, it is recommended to explore alternate Matlab software code such as faster RCNN to improve the accuracy of the system (Redmon and Farhadi 2016).

LIST OF REFERENCES

Ai-Ball. 2017. "Ai-Ball Specifications." Accessed August 16. http://www.thumbdrive.com/aiball/specs.html.

Beude, Dennis M. 2009. *The Engineering Design of Systems Models and Methods.* Hoboken, NJ: John Wiley & Sons.

Blackburn, M.R., R. T. Laird, H. R. Everett. 2001. *Unmanned Ground Vehicle (UGV) Lessons Learned.* San Diego, CA: Space and Naval Warfare Systems Center.

Bonin-Font, Francisco, Alberto Ortiz, and Gabriel Oliver. 2008. Visual Navigation for Mobile Robots: A Survey. *Journal of Intelligent & Robotic Systems* 53(264). doi: 10.1007/s10846-008-9235-4.

Bonnifait, Philippe, Jabbour, Maged and Cherfaoui, Véronique. 2008. "Autonomous Navigation in Urban Areas using GIS-Managed Information." *Int. J. of Vehicle Autonomous Systems,* 83–103.

D-Link. 2017. "D-Link: IP Camera." Accessed July 25, 2017. http://www.dlink.com.sg/product-category/products/surveillance/ip-camera/.

FLIR. 2017. "FLIR: Military & Defense.." Accessed August 03, 2017. http://www.dlink.com.sg/product-category/products/surveillance/ip-camera/.

Glenn, Russell W., and Gina Kingston. 2005. *Urban Battle Command in the 21st Century.* Santa Monica, CA: RAND.

Hanlon, Mike. 2005. "The Gladiator: U.S. Marines' Unmanned Ground Vehicle." New Atlas. August 27. http://newatlas.com/go/4484/.

John, Elza. 2017. "Matlab Expo 2017: Simplifying Image Processing and Computer Vision Application Development." MATLAB Expo. Accessed July 25, 2017. http://www.matlabexpo.com/in/2017/proceedings/simplifying-image-processing-computer-vision-application-development.pdf.

Keller, John. 2013. "Electro-Optical Sensor Payloads for Small UAVs." Ocotober 8. http://www.militaryaerospace.com/articles/print/volume-24/issue-10/technology-focus/electro-optical-sensor-payloads-for-small-uavs.html.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. "ImageNet Classification with Deep Neural Networks." *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 1097–1105.

Kruegle, Herman. 2011. *CCTV Surveillance: Video Practices and Technology*. Waltham, MA: Butterworth-Heinemann.

Logitech. 2017. “HD PRO WEBCAM C920.” Accessed July 24, 2017. http://www.logitech.com/en-us/product/hd-pro-webcam-c920.

Marr, Bernard. 2016. “4 Mind-Blowing Ways Facebook Uses Artificial Intelligence.” Accessed July 11, 2017. https://www.forbes.com/sites/bernardmarr/2016/12/29/4-amazing-ways-facebook-uses-deep-learning-to-learn-everything-about-you/2/#1d49afbb3090.

Mathworks. 2017a. “Deep Learning: 3 things you need to know.” Accessed July 20, 2017. https://www.mathworks.com/discovery/deep-learning.html.

———. 2017b. “Image Category Classification Using Deep Learning.” Accessed July 20, 2017. https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html.

———. 2017c. “Train a Cascade Object Detector” Accessed July 20, 2017. https://www.mathworks.com/help/vision/ug/train-a-cascade-object-detector.html

Mohri, Mehryar, Rostamizadeh, Afshin, and Talwalkar, Ameet. 2014. *Foundations of Machine Learning.* Cambirdge: MA: MIT Press.

Murphy, Kevin P. 2014. *Machine Learning: A Probabilistic Perpective*. Cambridge, MA: MIT Press.

National Research Council. 2002. *Technology Development for Army Unmanned Ground Vehicles.* Washington, DC: National Academies Press.

National Research Council, Division on Engineering and Physical Sciences, Naval Studies Board, Committee on Autonomous Vehicles in Support of Naval Operations. 2005. *Autonomous Vehicles in Support of Naval Operations.* National Academies Press.

Naval Studies Board, and National Research Council. 2005. *Autonomous Vehicles in Support of Naval Operation*s. Washington, DC: National Academies Press

Norvig, Stuart J. Russell and Peter. 1995. *Artificial Intelligence: A Modern Approach.* Upper Saddle River, NJ: Prentice Hall.

Redmon, Joseph, and Farhadi, Ali. 2016. “YOLO9000: Better, Faster, Stronger.” Deceber 25.

Schapire, Rob. 2008. “COS 511: Theoretical Machine Learning.” Accessed August 01, 2017. https://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf

Shin, Hoo Chang, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M. Summers. 2016. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning." *IEEE* 1.

Snider, Sara, and Mallory Simon. 2016. "How Robot, Explosives Took out Dallas Sniper in Unprecedented Way." Accessed July 24, 2017. http://www.cnn.com/2016/07/12/us/dallas-police-robot-c4-explosives/.

Vinciarelli, Allesandro, and Francesco Camastra. 2015. *Machine Learning for Audio, Image and Video Analysis.* London: Springer.

Winnefeld, James A., and Kendall, Frank. 2011. *Unmanned systems integrated roadmap FY 2011-2036.* Washington, DC: Office of the Secretary of Defense. US.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California